



PROGRAMMING: ARRAY

Dhruvin Dholiya



1. Write a C program to print all negative element from static array.

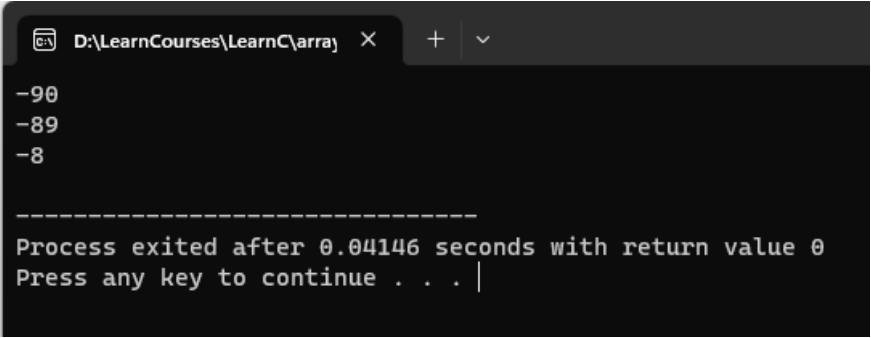
❖ Code:

```
#include<stdio.h>

int main() {
    int i, roll[5] = {-90, -89, 39, -8, 11};

    for (i = 0; i <= 4; i++) {
        if (roll[i] < 0) {
            printf("%d\n", roll[i]);
        }
    }
    return 0;
}
```

❖ Output:

A screenshot of a Windows command prompt window. The title bar shows the file path 'D:\LearnCourses\LearnC\array' and standard window controls. The terminal output displays the negative elements of the array: -90, -89, and -8, each on a new line. Below these, a separator line of dashes is shown, followed by the message 'Process exited after 0.04146 seconds with return value 0' and 'Press any key to continue . . . |' with a cursor.

```
D:\LearnCourses\LearnC\array >
-90
-89
-8

-----
Process exited after 0.04146 seconds with return value 0
Press any key to continue . . . |
```

2. Write a C program to print all negative element from dynamic array.

❖ Code:

```
#include<stdio.h>

int main() {
    int i, n;
    int arr[100];

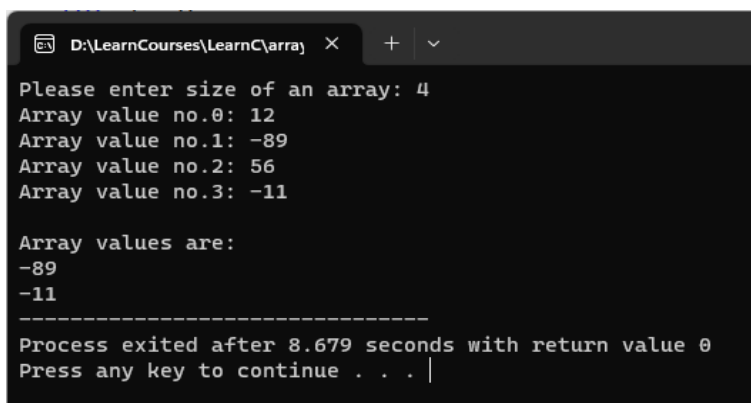
    printf("Please enter size of an array: ");
    scanf("%d", &n);

    for (i = 0; i < n; i++) {
        printf("Array value no.%d: ", i);
        scanf("%d", &arr[i]);
    }

    printf("\nArray values are: ");
    for (i = 0; i < n; i++) {
        if (arr[i] < 0) {
            printf("\n%d", arr[i]);
        }
    }

    return 0;
}
```

❖ Output:



```
D:\LearnCourses\LearnC\array X + v
Please enter size of an array: 4
Array value no.0: 12
Array value no.1: -89
Array value no.2: 56
Array value no.3: -11

Array values are:
-89
-11
-----
Process exited after 8.679 seconds with return value 0
Press any key to continue . . . |
```

3. Write a C program to print all even element from static array.

❖ Code:

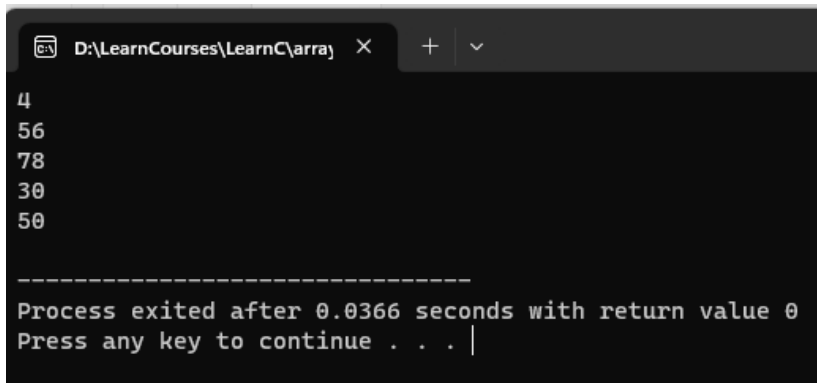
```
#include<stdio.h>

int main() {

    int i, num[10] = { 23, 4, 67, 56, 78, 45, 3, 30, 89, 50};

    for (i = 0; i <= 10; i++) {
        if (num[i] % 2 == 0) {
            printf("%d\n", num[i]);
        }
    }
    return 0;
}
```

❖ Output:

A screenshot of a Windows command prompt window. The title bar shows the file path 'D:\LearnCourses\LearnC\array'. The output of the program is displayed in the console, showing the even numbers from the array: 4, 56, 78, 30, and 50, each on a new line. Below the numbers, there is a separator line of dashes, followed by the message 'Process exited after 0.0366 seconds with return value 0' and 'Press any key to continue . . . |' with a cursor.

```
D:\LearnCourses\LearnC\array >
4
56
78
30
50

-----
Process exited after 0.0366 seconds with return value 0
Press any key to continue . . . |
```

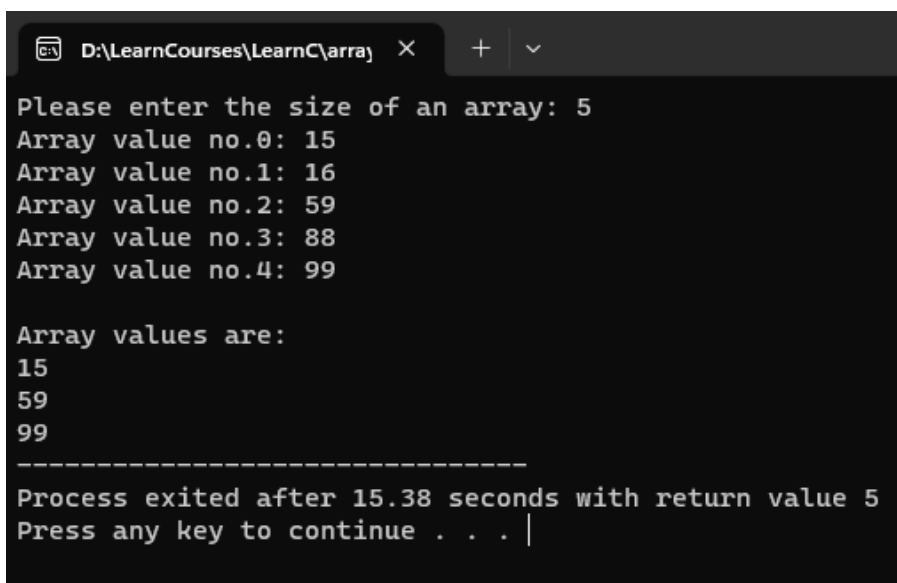
4. Write a C program to print all odd element from dynamic array.

❖ Code:

```
#include<stdio.h>

int main () {
    int i, n;
    int arr[100];
    printf("Please enter the size of an array: ");
    scanf("%d", &n);
    for (i = 0; i < n; i++) {
        printf("Array value no.%d: ", i);
        scanf("%d", &arr[i]);
    }
    printf("\nArray values are: ");
    for (i = 0; i < n; i++) {
        if (arr[i] % 2 == 1) {
            printf("\n%d", arr[i]);
        }
    }
}
```

❖ Output:



```
D:\LearnCourses\LearnC\array X + v
Please enter the size of an array: 5
Array value no.0: 15
Array value no.1: 16
Array value no.2: 59
Array value no.3: 88
Array value no.4: 99

Array values are:
15
59
99
-----
Process exited after 15.38 seconds with return value 5
Press any key to continue . . . |
```

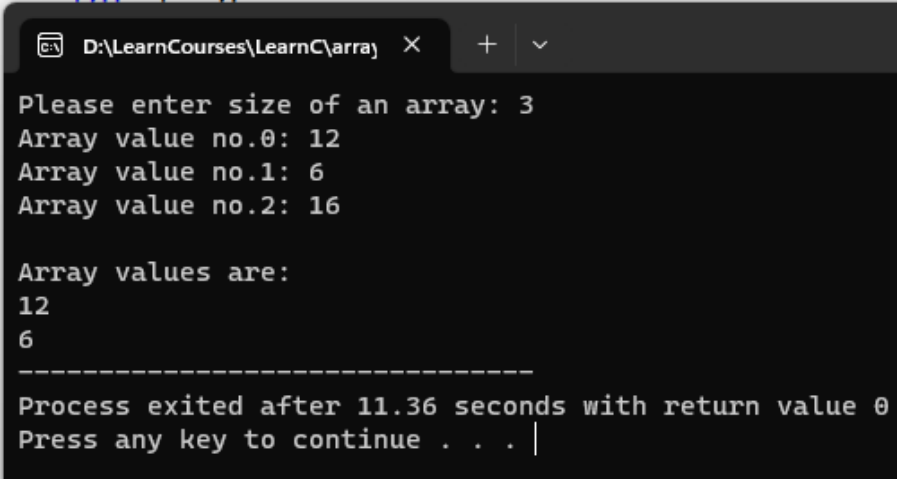
5. Write a C program to print all element that are divisible by 3 from dynamic array.

❖ Code:

```
#include<stdio.h>

int main() {
    int i, n;
    int arr[100];
    printf("Please enter size of an array: ");
    scanf("%d", &n);
    for (i = 0; i < n; i++) {
        printf("Array value no.%d: ", i);
        scanf("%d", &arr[i]);
    }
    printf("\nArray values are: ");
    for (i = 0; i < n; i++) {
        if (arr[i] % 3 == 0) {
            printf("\n%d", arr[i]);
        }
    }
    return 0;
}
```

❖ Output:

A screenshot of a terminal window with a dark background. The window title bar shows the file path 'D:\LearnCourses\LearnC\array' and standard window controls. The terminal output shows the program's execution: it prompts for the array size (3), then prompts for three array elements (12, 6, 16). It then prints the array values and lists the elements divisible by 3 (12 and 6). The output ends with a separator line, the process exit message, and a prompt to press any key to continue.

```
D:\LearnCourses\LearnC\array  X  +  v

Please enter size of an array: 3
Array value no.0: 12
Array value no.1: 6
Array value no.2: 16

Array values are:
12
6
-----
Process exited after 11.36 seconds with return value 0
Press any key to continue . . . |
```

6. Write a C program to find max element from an dynamic array.

❖ Code:

```
#include <stdio.h>

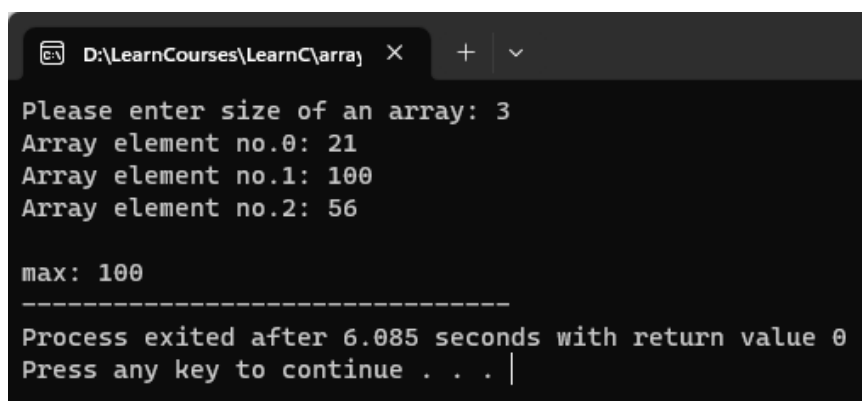
int main() {
    int i, n, arr[i], max=0;
    printf("Please enter size of an array: ");
    scanf("%d", &n);

    for (i = 0; i < n; i++) {
        printf("Array element no.%d: ", i);
        scanf("%d", &arr[i]);
    }

    for (i = 0; i < n; i++) {
        if (arr[i] > max) {
            max = arr[i];
        }
    }

    printf("\nmax: %d", max);
    return 0;
}
```

❖ Output:



```
D:\LearnCourses\LearnC\array x + v
Please enter size of an array: 3
Array element no.0: 21
Array element no.1: 100
Array element no.2: 56

max: 100
-----
Process exited after 6.085 seconds with return value 0
Press any key to continue . . . |
```

7. Write C program to find second largest number in array.

❖ Code:

```
#include <stdio.h>
#define MAX_SIZE 1000

int main() {
    int i, n, arr[MAX_SIZE], temp=0, max2=0;

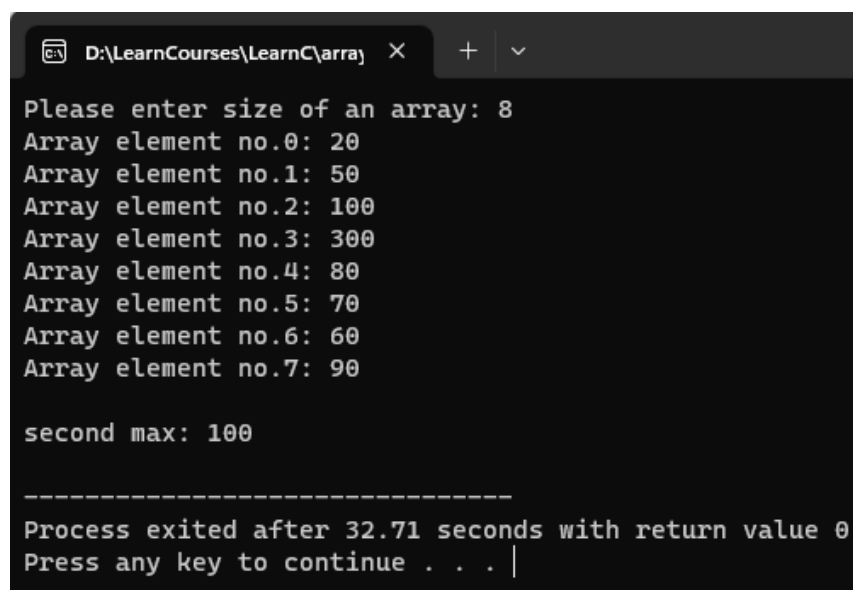
    printf("Please enter size of an array: ");
    scanf("%d", &n);

    for (i = 0; i < n; i++) {
        printf("Array element no.%d: ", i);
        scanf("%d", &arr[i]);
    }

    for (i = 0; i < n; i++) {
        if (arr[i] > temp) {
            max2 = temp;
            temp = arr[i];
        } else if (arr[i] > max2 && arr[i] < temp) {
            arr[i] = max2;
        }
    }
    printf("\nsecond max: %d\n", max2);

    return 0;
}
```

❖ Output:

A screenshot of a Windows command prompt window with a dark background. The title bar shows the file path 'D:\LearnCourses\LearnC\array'. The program prompts the user to enter the size of an array, which is 8. It then prompts for 8 array elements: 20, 50, 100, 300, 80, 70, 60, and 90. The output shows the second maximum value is 100. At the bottom, it says 'Process exited after 32.71 seconds with return value 0' and 'Press any key to continue . . . |'.

```
D:\LearnCourses\LearnC\array x + v
Please enter size of an array: 8
Array element no.0: 20
Array element no.1: 50
Array element no.2: 100
Array element no.3: 300
Array element no.4: 80
Array element no.5: 70
Array element no.6: 60
Array element no.7: 90

second max: 100

-----
Process exited after 32.71 seconds with return value 0
Press any key to continue . . . |
```


8. Write C program to Update the element into array.

❖ Code:

```
#include <stdio.h>
int main() {
    int i, n, arr[i], ind, newEle;

    printf("Please enter size of an array: ");
    scanf("%d", &n);

    for (i = 0; i < n; i++) {
        printf("Array element no.%d: ", i);
        scanf("%d", &arr[i]);
    }

    printf("\nWhich element do you want to change?: ");
    scanf("%d", &ind);

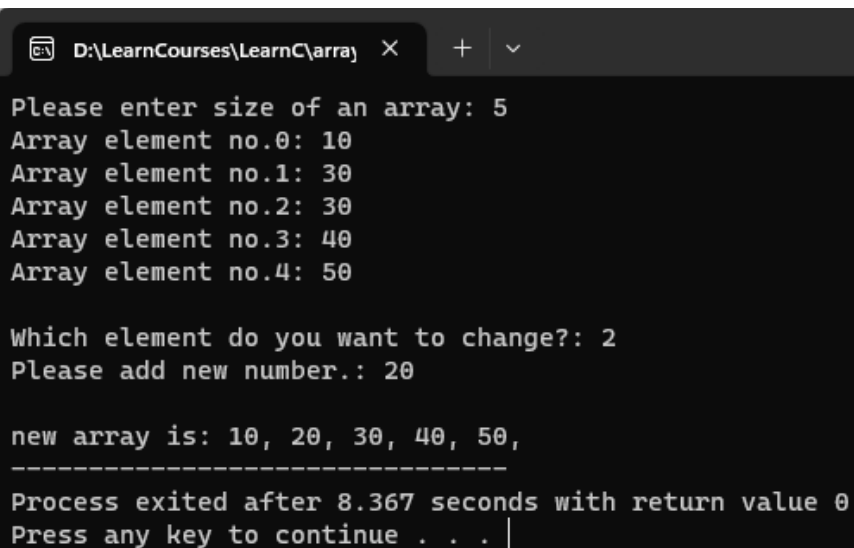
    printf("Please add new number.: ");
    scanf("%d", &newEle);

    arr[ind - 1] = newEle;

    printf("\nnew array is: ");

    for(i = 0; i < n; i++) {
        printf("%d, ", arr[i]);
    }
    return 0;
}
```

❖ Output:

A screenshot of a Windows command prompt window with a dark background. The title bar shows the file path 'D:\LearnCourses\LearnC\array'. The program's output is displayed in white text. It prompts the user to enter the size of an array (5), then lists the elements of the array (10, 30, 30, 40, 50). It then asks which element to change (2), prompts for a new number (20), and finally displays the updated array (10, 20, 30, 40, 50). The window ends with a separator line and a message about the process exiting and a prompt to press any key to continue.

```
D:\LearnCourses\LearnC\array X + v
Please enter size of an array: 5
Array element no.0: 10
Array element no.1: 30
Array element no.2: 30
Array element no.3: 40
Array element no.4: 50

Which element do you want to change?: 2
Please add new number.: 20

new array is: 10, 20, 30, 40, 50,
-----
Process exited after 8.367 seconds with return value 0
Press any key to continue . . . |
```

9. Write C program to Insert the element into array.

❖ Code:

```
#include <stdio.h>
int main() {
    int i, n, arr[i], ind, newEle;

    printf("Please enter size of an array: ");
    scanf("%d", &n);

    for (i = 0; i < n; i++) {
        printf("Array ele no.%d: ", i);
        scanf("%d", &arr[i]);
    }

    printf("Where do you want to add element?: ");
    scanf("%d", &ind);
    printf("Please add new element.: ");
    scanf("%d", &newEle);

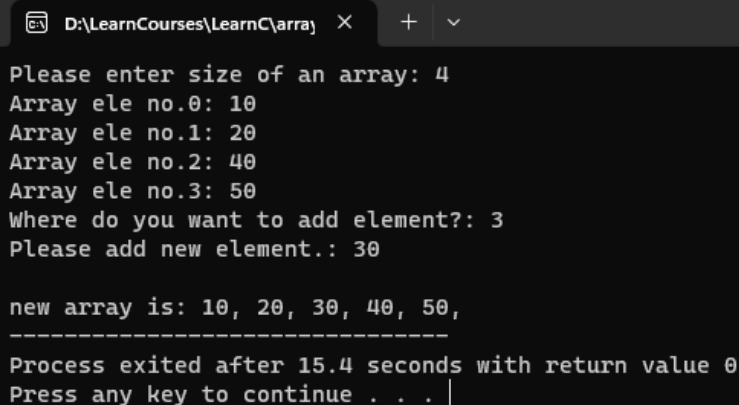
    n++;

    for (i = n - 1; i >= ind; i--) {
        arr[i] = arr[i - 1];
    }

    arr[ind - 1] = newEle;

    printf("\nnew array is: ");
    for(i = 0; i < n; i++) {
        printf("%d, ", arr[i]);
    }
    return 0;
}
```

❖ Output:



The screenshot shows a Windows command prompt window with the file path 'D:\LearnCourses\LearnC\array'. The program prompts the user to enter the size of an array (4), then enters four elements (10, 20, 40, 50). It then prompts for the index to insert a new element (3) and the new element value (30). The final output shows the new array: 10, 20, 30, 40, 50. The process exited after 15.4 seconds with a return value of 0.

```
D:\LearnCourses\LearnC\array X + v
Please enter size of an array: 4
Array ele no.0: 10
Array ele no.1: 20
Array ele no.2: 40
Array ele no.3: 50
Where do you want to add element?: 3
Please add new element.: 30

new array is: 10, 20, 30, 40, 50,
-----
Process exited after 15.4 seconds with return value 0
Press any key to continue . . . |
```

10. Write C program to Delete the element into array.

❖ Code:

```
#include <stdio.h>
int main() {
    int i, n, arr[100], ind;

    printf("Please enter size of an array: ");
    scanf("%d", &n);

    for (i = 0; i < n; i++) {
        printf("Array ele no.%d: ", i);
        scanf("%d", &arr[i]);
    }

    printf("Which element do you want to remove?: ");
    scanf("%d", &ind);

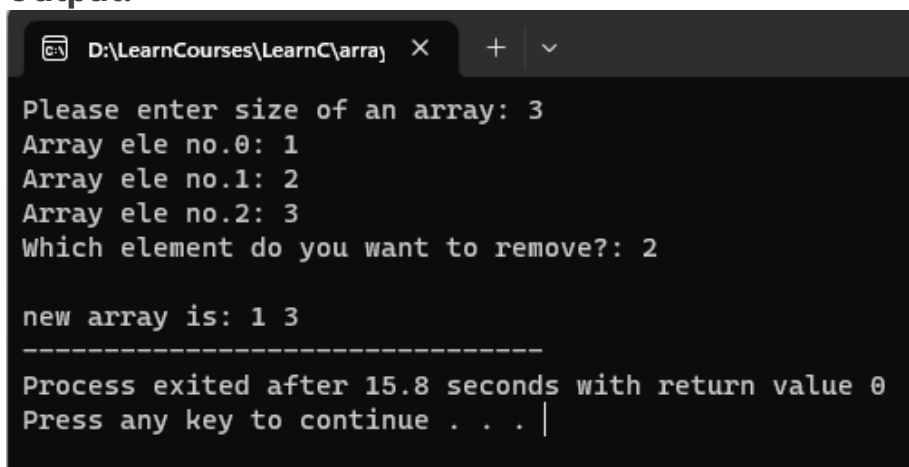
    n--;

    for (i = ind - 1; i <= n; i++) {
        arr[i] = arr[i + 1];
    }

    printf("\nnew array is: ");
    for(i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }

    return 0;
}
```

❖ Output:

A screenshot of a terminal window showing the execution of a C program. The window title is "D:\LearnCourses\LearnC\array". The output text is as follows:

```
Please enter size of an array: 3
Array ele no.0: 1
Array ele no.1: 2
Array ele no.2: 3
Which element do you want to remove?: 2

new array is: 1 3
-----
Process exited after 15.8 seconds with return value 0
Press any key to continue . . . |
```

11. Write C program to left rotate an array element.

❖ Code:

```
#include <stdio.h>

int main() {
    int i, n, arr[100], first;

    printf("Please enter size of an array: ");
    scanf("%d", &n);

    for (i = 0; i < n; i++) {
        printf("Please enter element no.%d: ", i);
        scanf("%d", &arr[i]);
    }

    first = arr[0];

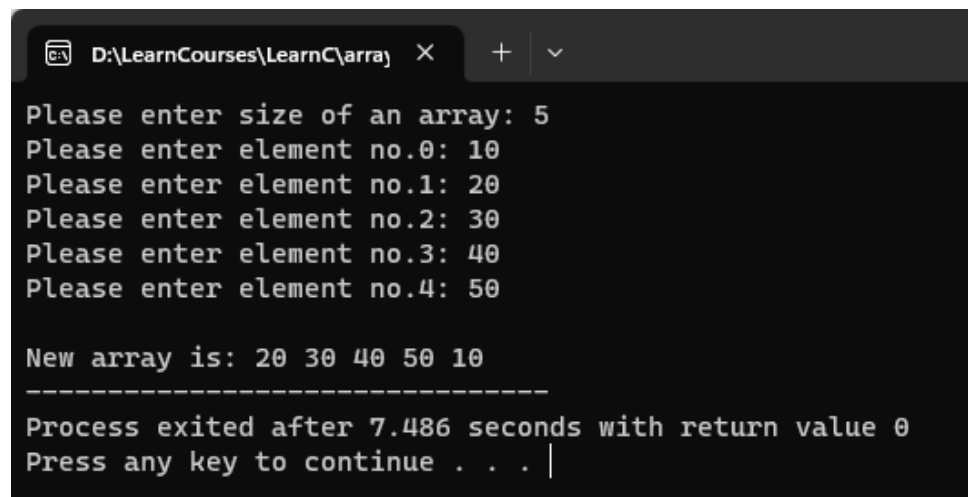
    for (i = 0; i < n; i++) {
        arr[i] = arr[i + 1];
    }

    arr[n - 1] = first;

    printf("\nNew array is: ");
    for (i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }

    return 0;
}
```

❖ Output:



The screenshot shows a Windows command prompt window with the title bar "D:\LearnCourses\LearnC\array". The program prompts the user to enter the size of an array (5) and then five elements (10, 20, 30, 40, 50). It then displays the new array after left rotation: 20 30 40 50 10. The window also shows the process exit time and a prompt to press any key to continue.

```
D:\LearnCourses\LearnC\array  X  +  v

Please enter size of an array: 5
Please enter element no.0: 10
Please enter element no.1: 20
Please enter element no.2: 30
Please enter element no.3: 40
Please enter element no.4: 50

New array is: 20 30 40 50 10
-----
Process exited after 7.486 seconds with return value 0
Press any key to continue . . . |
```

12. Write C program to right rotate an array element.

❖ Code:

```
#include <stdio.h>

int main() {
    int i, n, arr[100], last;
    printf("Please enter size of an array: ");
    scanf("%d", &n);

    for (i = 0; i < n; i++) {
        printf("Please enter element no.%d: ", i);
        scanf("%d", &arr[i]);
    }

    last = arr[n - 1];

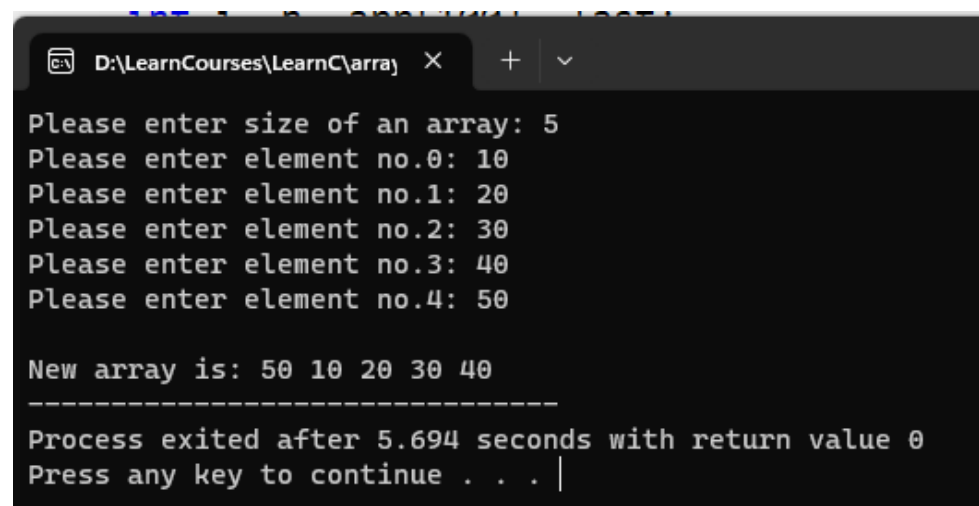
    for (i = n - 1; i >= 0; i--) {
        arr[i] = arr[i - 1];
    }

    arr[0] = last;

    printf("\nNew array is: ");
    for (i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }

    return 0;
}
```

❖ Output:

A screenshot of a Windows command prompt window with a dark background. The title bar shows the file path 'D:\LearnCourses\LearnC\array'. The program prompts the user to enter the size of an array (5) and then five elements (10, 20, 30, 40, 50). It then displays the new array after a right rotation: 50 10 20 30 40. The window also shows the execution time (5.694 seconds) and the return value (0).

```
D:\LearnCourses\LearnC\array >
Please enter size of an array: 5
Please enter element no.0: 10
Please enter element no.1: 20
Please enter element no.2: 30
Please enter element no.3: 40
Please enter element no.4: 50

New array is: 50 10 20 30 40
-----
Process exited after 5.694 seconds with return value 0
Press any key to continue . . . |
```

13. Write C program to addition of two matrices.

❖ Code:

```
#include<stdio.h>

int main() {
    int arr1[3][3] = {
        {10, 20, 30},
        {30, 50, 60},
        {70, 80, 90}
    };

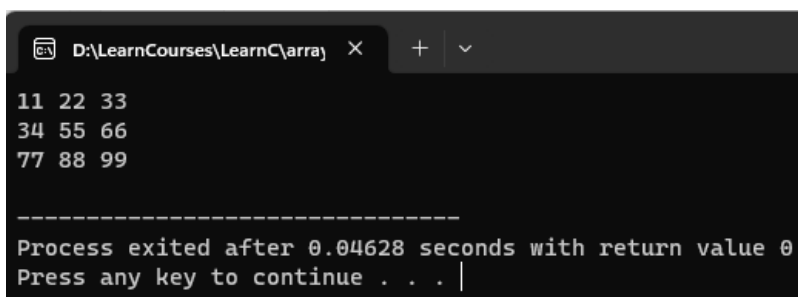
    int arr2[3][3] = {
        {1, 2, 3},
        {4, 5, 6},
        {7, 8, 9}
    };

    int i, x;
    int res[3][3] = {0};

    for (i = 0; i < 3; i++) {
        for(x = 0; x < 3; x++) {
            res[i][x] = arr1[i][x] + arr2[i][x];
        }
    }

    for (i = 0; i < 3; i++) {
        for(x = 0; x < 3; x++) {
            printf("%d ", res[i][x]);
        }
        printf("\n");
    }
    return 0;
}
```

❖ Output:

A screenshot of a terminal window showing the output of a C program. The window title is "D:\LearnCourses\LearnC\array". The output displays the addition of two 3x3 matrices. The first matrix has values 10, 20, 30 in the first row; 30, 50, 60 in the second row; and 70, 80, 90 in the third row. The second matrix has values 1, 2, 3 in the first row; 4, 5, 6 in the second row; and 7, 8, 9 in the third row. The resulting matrix, shown in the output, has values 11, 22, 33 in the first row; 34, 55, 66 in the second row; and 77, 88, 99 in the third row. Below the output, a separator line is shown, followed by the message "Process exited after 0.04628 seconds with return value 0" and "Press any key to continue . . . |".

```
D:\LearnCourses\LearnC\array X + v
11 22 33
34 55 66
77 88 99

-----
Process exited after 0.04628 seconds with return value 0
Press any key to continue . . . |
```

14. Write C program matrix convert into transpose matrix.

❖ Code:

```
#include<stdio.h>

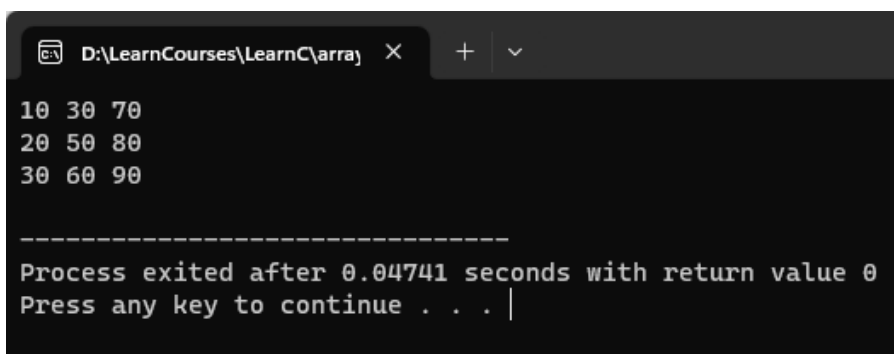
int main() {
    int arr[3][3] = {
        {10, 20, 30},
        {30, 50, 60},
        {70, 80, 90}
    };

    int i, x, trans[3][3] = {0};

    for (i = 0; i < 3; i++) {
        for(x = 0; x < 3; x++) {
            trans[x][i] = arr[i][x];
        }
    }

    for (i = 0; i < 3; i++) {
        for(x = 0; x < 3; x++) {
            printf("%d ", trans[i][x]);
        }
        printf("\n");
    }
    return 0;
}
```

❖ Output:



```
D:\LearnCourses\LearnC\array  ×  +  ▾

10 30 70
20 50 80
30 60 90

-----
Process exited after 0.04741 seconds with return value 0
Press any key to continue . . . |
```

15. Write C program to find sum of diagonal elements of a matrix.

❖ Code:

```
#include<stdio.h>

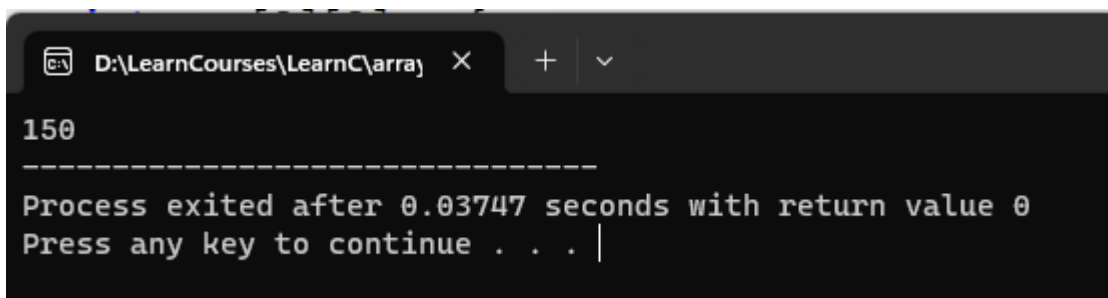
int main() {
    int arr[3][3] = {
        {10, 20, 30},
        {30, 50, 60},
        {70, 80, 90}
    };

    int i, x, sum = 0;

    for (i = 0; i < 3; i++) {
        for(x = 0; x < 3; x++) {
            if(i == x) {
                sum = sum + arr[i][x];
            }
        }
    }

    printf("%d", sum);
    return 0;
}
```

❖ Output:



```
D:\LearnCourses\LearnC\array X + v
150
-----
Process exited after 0.03747 seconds with return value 0
Press any key to continue . . . |
```


16. Write a C program to sum of all even element from an array.

❖ Code:

```
#include <stdio.h>

int main() {
    int i, n, arr[i], sum=0;

    printf("Please enter size of an array: ");
    scanf("%d", &n);

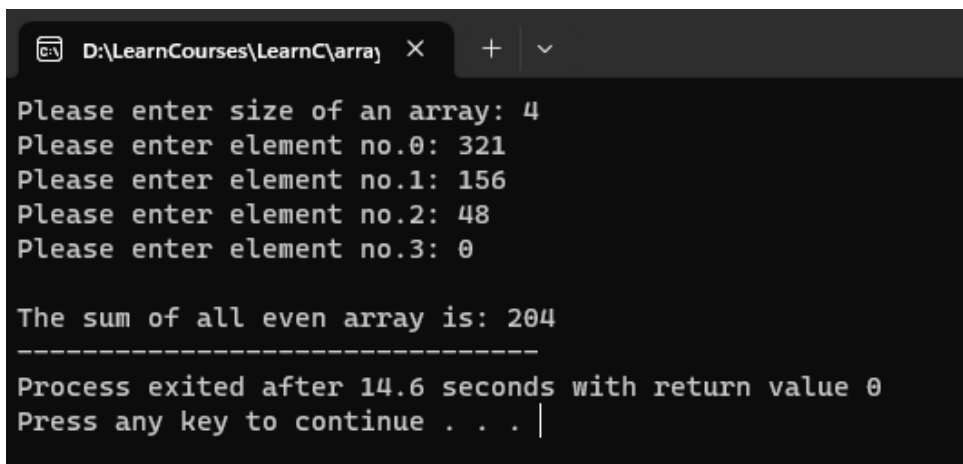
    for (i = 0; i < n; i++) {
        printf("Please enter element no.%d: ", i);
        scanf("%d", &arr[i]);
    }

    for(i = 0; i < n; i++) {
        if(arr[i] % 2 == 0) {
            sum = sum + arr[i];
        }
    }

    printf("\nThe sum of all even array is: %d", sum);

    return 0;
}
```

❖ Output:



```
D:\LearnCourses\LearnC\array X + v
Please enter size of an array: 4
Please enter element no.0: 321
Please enter element no.1: 156
Please enter element no.2: 48
Please enter element no.3: 0

The sum of all even array is: 204
-----
Process exited after 14.6 seconds with return value 0
Press any key to continue . . . |
```

17. Write a C program to find average of an element from an array.

❖ Code:

```
#include <stdio.h>

int main() {
    int i, n, arr[i], sum=0, avg=0;

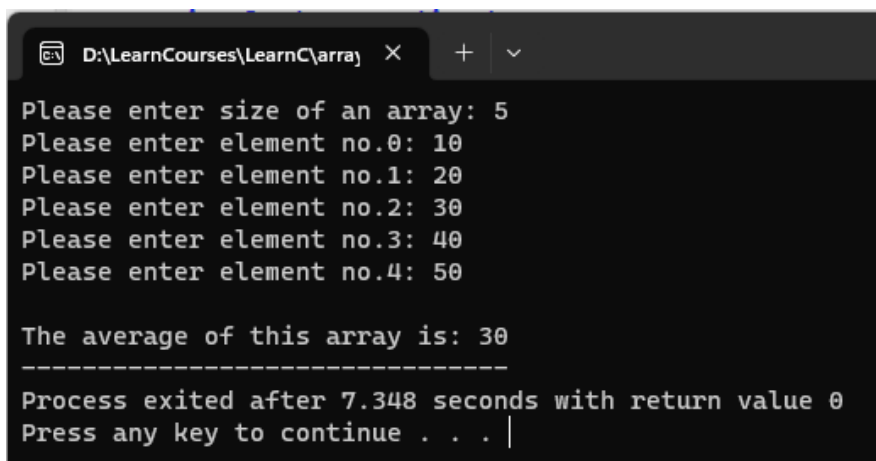
    printf("Please enter size of an array: ");
    scanf("%d", &n);

    for (i = 0; i < n; i++) {
        printf("Please enter element no.%d: ", i);
        scanf("%d", &arr[i]);
    }

    for(i = 0; i < n; i++) {
        sum = sum + arr[i];
        avg = sum / n;
    }

    printf("\nThe average of this array is: %d", avg);
    return 0;
}
```

❖ Output:



The screenshot shows a terminal window with the following output:

```
D:\LearnCourses\LearnC\array x + v
Please enter size of an array: 5
Please enter element no.0: 10
Please enter element no.1: 20
Please enter element no.2: 30
Please enter element no.3: 40
Please enter element no.4: 50

The average of this array is: 30
-----
Process exited after 7.348 seconds with return value 0
Press any key to continue . . . |
```

18. Write a C program to count number of students in each group (0-9, 10- 19, 20-29 90-99, 100-100) for the given students marks.

Marks: 85, 66, 37, 45, 68, 23, 99, 100, 81, 70, 42, 55, 68, 77, 96, 18

❖ **Code:**

```
#include <stdio.h>

int main() {

    int marks[100], i, n, group, count[11] = {0};

    printf("How many students: ");

    scanf("%d", &n);

    for (i = 0; i < n; i++) {

        printf("Please enter marks of student's no.%d: ", i + 1);

        scanf("%d", &marks[i]);

    }

    for (i=0; i<n; i++) {

        group = marks[i] / 10;

        count[group]++;

    }

    printf("\nGroup\tNumber of Students\n");

    for (i=0; i<10; i++) {

        printf("%d-%d\t%d\n", i * 10, i * 10 + 9, count[i]);

    }

    printf("100\t%d\n", count[10]);

    return 0;

}
```

❖ Output:

```
D:\LearnCourses\LearnC\array X + v
How many students: 15
Please enter marks of student's no.1: 0
Please enter marks of student's no.2: 9
Please enter marks of student's no.3: 15
Please enter marks of student's no.4: 27
Please enter marks of student's no.5: 44
Please enter marks of student's no.6: 36
Please enter marks of student's no.7: 99
Please enter marks of student's no.8: 100
Please enter marks of student's no.9: 100
Please enter marks of student's no.10: 59
Please enter marks of student's no.11: 67
Please enter marks of student's no.12: 88
Please enter marks of student's no.13: 88
Please enter marks of student's no.14: 88
Please enter marks of student's no.15: 50

Group    Number of Students
0-9      2
10-19    1
20-29    1
30-39    1
40-49    1
50-59    2
60-69    1
70-79    0
80-89    3
90-99    1
100      2

-----
Process exited after 44.4 seconds with return value 0
Press any key to continue . . . |
```